Scalable Gaussian Process Classification via Expectation Propagation

Daniel Hernández–Lobato[1],

May 21, 2015

joint work with

José Miguel Hernández-Lobato[2]

[1]Universidad Autónoma de Madrid.
[2]Harvard University.

Introduction

Under binary Gaussian Process classification one assumes that $y_i = \text{sign}(f(\mathbf{x}_i) + \epsilon_i)$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $y_i \in \{1, -1\}$.

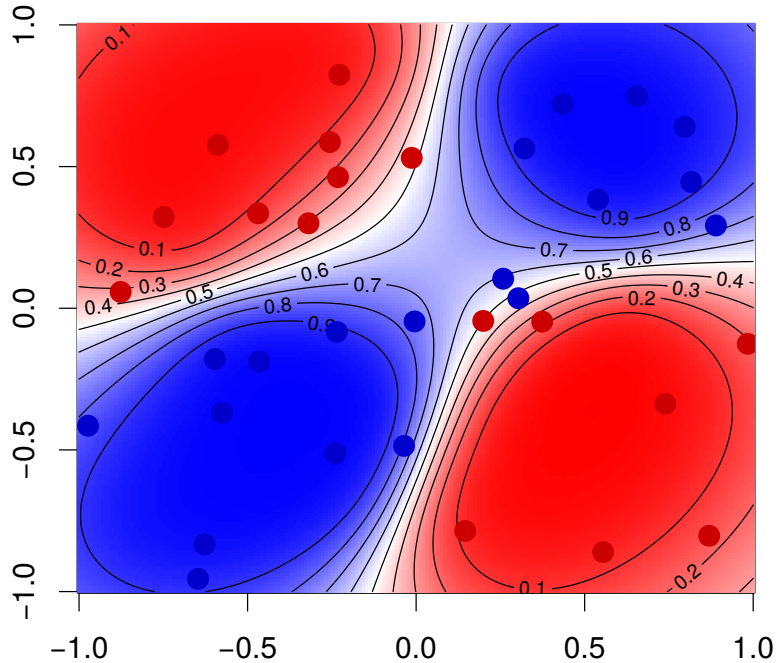
$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)), \text{ for some covariance function } k.$$

Learning uses Bayes rule:

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y})}.$$

Intractable in general!

Approximated in practice by VB, EP or Laplace's method.



Requires inverting **a matrix of size** $n \times n$ and scales like $\mathcal{O}(n^3)$.

(Rasmussen & Williams, 2006)

Sparse Gaussian Process Classification

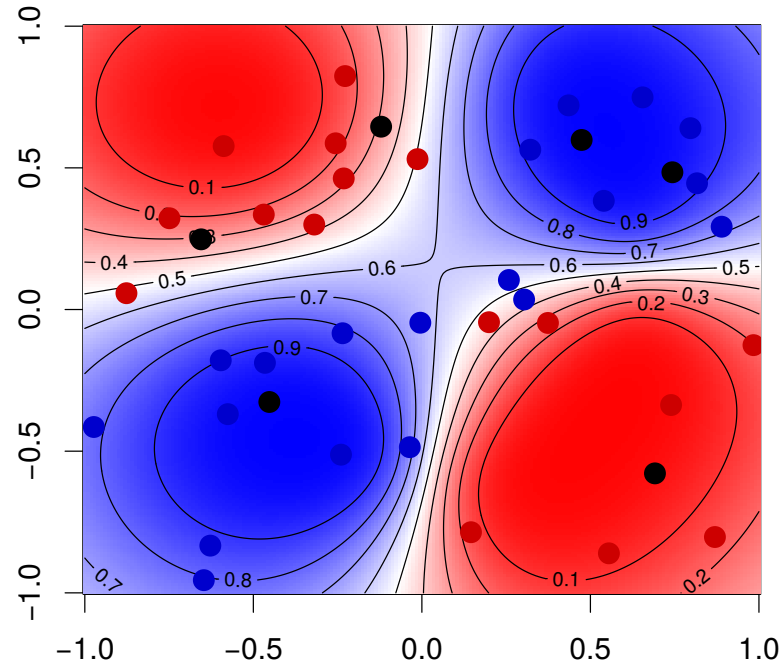
Reduces the training time to $\mathcal{O}(nm^2)$, where $m \ll n$.

A popular approach introduces m pseudoinputs with associated values that are **marginalized** (Naish-Guzman & Holden, 2008).

Given $\overline{\mathbf{X}}$ pseudoinputs, let $\overline{\mathbf{f}}$ be the associated functional values:

$$
\begin{aligned}
p(\mathbf{f}) &= \int p(\mathbf{f}|\overline{\mathbf{f}}, \overline{\mathbf{X}}) p(\overline{\mathbf{f}}|\overline{\mathbf{X}}) d\overline{\mathbf{f}} \\
&\approx \int \left[\prod_{i=1}^{m} p(f(\mathbf{x}_i)|\overline{\mathbf{f}}, \overline{\mathbf{X}}) \right] p(\overline{\mathbf{f}}|\overline{\mathbf{X}}) d\overline{\mathbf{f}} \\
&= p_{\text{FITC}}(\mathbf{f}|\overline{\mathbf{X}})
\end{aligned}
$$

(Quiñonero Candela & Rasmussen, 2005)



The covariance matrix of the FITC prior has a low-rank form which allows to handle datasets with a few **thousand instances**.

Scalable Variational Gaussian Process Classification

Allows to handle datasets with **millions** of data instances.

Combines ideas from sparse variational Gaussian processes (Titsias, 2009) and stochastic variational inference (Hoffman et al., 2013).

The key is **not marginalize** the values $\overline{\mathbf{f}}$ of the pseudoinputs.

The approximation to $p(\mathbf{f}|\mathbf{y})$ is set to be $q(\mathbf{f}) = \int p(\mathbf{f}|\overline{\mathbf{f}})q(\overline{\mathbf{f}})d\overline{\mathbf{f}}$, were $q(\overline{\mathbf{f}})$ is an inferred Gaussian distribution that approximates $p(\overline{\mathbf{f}}|\mathbf{y})$.

The parameters of $q(\overline{\mathbf{f}})$ are found in practice by maximizing:

$$\log p(\mathbf{y}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \left[\log p(y_i|f_i)\right] - \mathrm{KL}(q(\overline{\mathbf{f}})||p(\overline{\mathbf{f}}|\overline{\mathbf{X}}))$$

whose gradient has a sum over the instances (Hensman et al., 2015).

The cost in training time is $\mathcal{O}(m^3)$, but requires **quadratures**.

Scalable Gaussian Process Classification via EP I

We compute $q(\bar{\mathbf{f}})$ by **approximately minimizing** $\mathrm{KL}(p(\bar{\mathbf{f}}|\mathbf{y})|q(\bar{\mathbf{f}}))$ instead of $\mathrm{KL}(q(\bar{\mathbf{f}})|p(\bar{\mathbf{f}}|\mathbf{y}))$ using expectation propagation (EP).

We also assume that $p(\mathbf{f}|\mathbf{y}) \approx \int p(\mathbf{f}|\bar{\mathbf{f}})q(\bar{\mathbf{f}})d\bar{\mathbf{f}}$. The exact posterior is:

$$p(\bar{\mathbf{f}}|\mathbf{y}) = \frac{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\bar{\mathbf{f}})d\mathbf{f}p(\bar{\mathbf{f}}|\overline{\mathbf{X}})}{p(\mathbf{y})} \approx \frac{\left[\prod_{i=1}^{n}\int p(y_i|f_i)p(f_i|\bar{\mathbf{f}})df_i\right]p(\bar{\mathbf{f}}|\overline{\mathbf{X}})}{p(\mathbf{y})}$$

where we have used the FITC approximation (Qi et al., 2010)[3].

The only **non-Gaussian** factors are those of the likelihood.

$$\overbrace{\int p(y_i|f_i)p(f_i|\bar{\mathbf{f}})df_i}^{\text{Exact Factor } \phi_i(\bar{\mathbf{f}})} \quad \xrightarrow{\text{EP Proj.}} \quad \overbrace{\tilde{s}_i \exp\left\{-\frac{\tilde{\nu}_i}{2}\bar{\mathbf{f}}^{\mathrm{T}}\tilde{\boldsymbol{\upsilon}}_i\tilde{\boldsymbol{\upsilon}}_i^{\mathrm{T}}\bar{\mathbf{f}} + \tilde{\mu}_i\bar{\mathbf{f}}^{\mathrm{T}}\tilde{\boldsymbol{\upsilon}}_i\right\}}^{\text{Approximate Gaussian Factor } \tilde{\phi}_i(\bar{\mathbf{f}})}$$

where $\tilde{\boldsymbol{\upsilon}}_i = \mathbf{K}_{\bar{\mathbf{f}},\bar{\mathbf{f}}}^{-1}\mathbf{K}_{\bar{\mathbf{f}},f_i}$, *i.e.*, only $\mathcal{O}(m)$ parameters **are stored**.

[3] The variational approach can also be derived in this way.

Scalable Gaussian Process Classification via EP II

$$q(\overline{\mathbf{f}}) = \frac{1}{Z_q} \prod_{i=1}^{n} \tilde{\phi}_i(\overline{\mathbf{f}}) p(\overline{\mathbf{f}}|\overline{\mathbf{X}}) \,.$$

Let $q^{\setminus i} \propto q/\tilde{\phi}_i$. Each approximate factor $\tilde{\phi}_i$ is updated to **minimize** $\mathrm{KL}(Z_i^{-1}\phi_i q^{\setminus i}||q)$, which involves matching moments.

Z_i can be computed in **closed** form. Moments obtained from its derivatives with respect to the natural parameters or $q^{\setminus i}$.

After convergence, $Z_q \approx p(\mathbf{y})$ and its gradients w.r.t. ξ_j are:

$$\log Z_q = \Phi(\theta) - \Phi(\theta_{\text{prior}}) + \sum_{i=1}^{n} \log \tilde{s}_i \,, \quad \log \tilde{s}_i = \log Z_i + \Phi(\theta^{\setminus i}) - \Phi(\theta) \,,$$

$$\frac{\partial \log Z_q}{\partial \xi_j} = \eta^{\mathrm{T}} \frac{\partial \theta_{\text{prior}}}{\partial \xi_j} - \eta_{\text{prior}}^{\mathrm{T}} \frac{\partial \theta_{\text{prior}}}{\partial \xi_j} + \sum_{i=1}^{n} \frac{\partial \log Z_i}{\partial \xi_j}$$

with θ, θ_{prior} and $\theta^{\setminus i}$ the **natural parameters** of q, $p(\overline{\mathbf{f}}|\overline{\mathbf{X}})$ and $q^{\setminus i}$, and η, η_{prior} the corresponding **expected sufficient statistics**.

Extends the results of (Seeger, 2006).

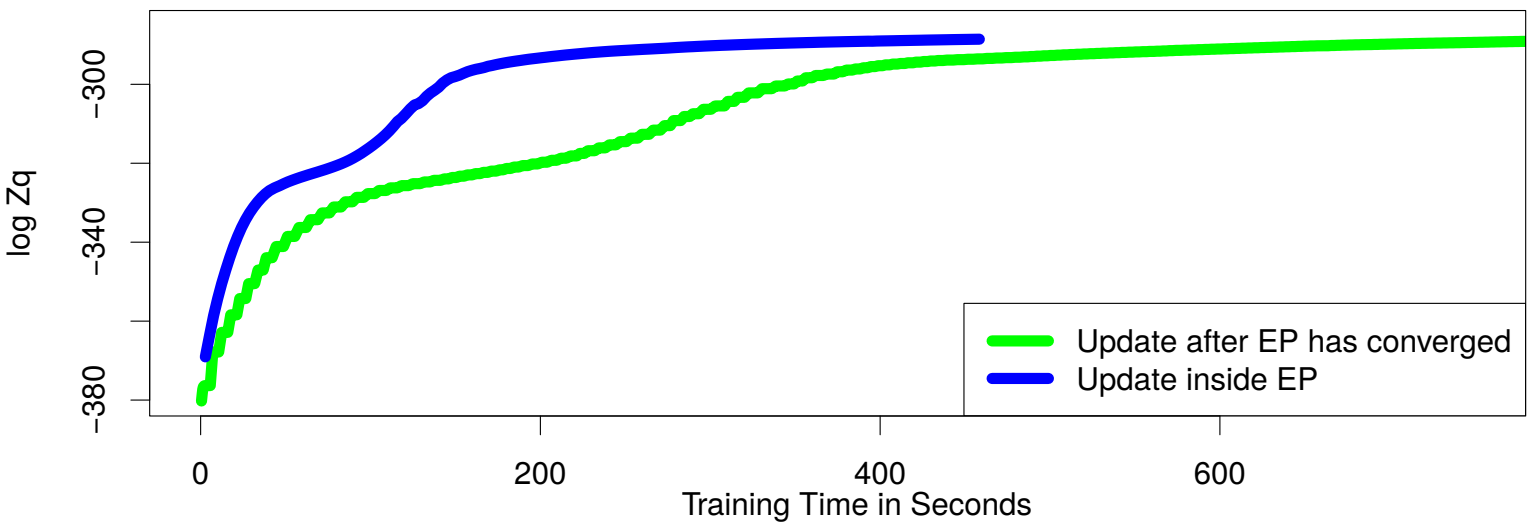
Speeding Up Hyper-parameter Estimation in EP

Waiting until EP converges is a waste of time!

We update all hyper-parameters ξ_j after a parallel update of each $\tilde{\phi}_i$.

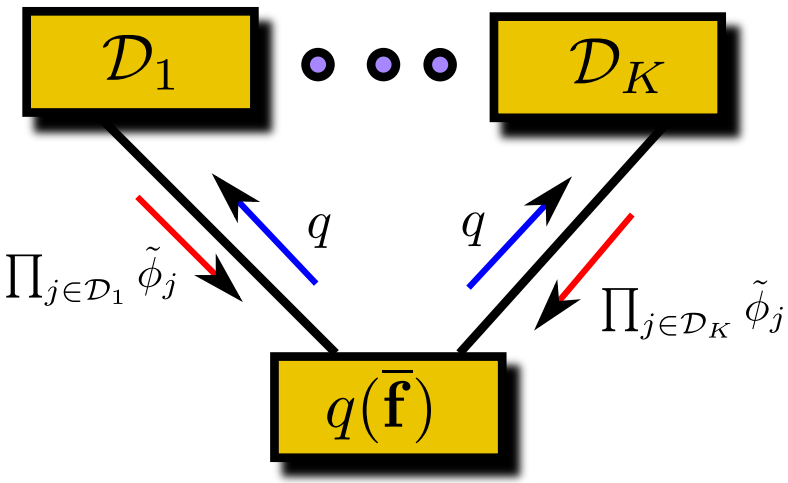
Intuitive interpretation:

1. At convergence we obtain a stationary point of Z_q (Minka, 2001).
2. The EP updates can be understood as natural gradient descent in Z_q assuming all other $\tilde{\phi}_i$ remain fixed (Heskes & Zoeter, 2002).
3. An inner update of each ξ_j using gradient ascent, assuming each $\tilde{\phi}_i$ is fixed, is also expected to be effective for maximizing Z_q.



Distributed EP updates and Gradient Computation

Use ideas from (Gelman et al., 2014) to run EP in a distributed way.



1. We split the data across K nodes.
2. A master node sends q to each other node.
3. At each node we compute at once each $q^{\setminus i}$ and each approximate factor $\tilde{\phi}_i$.
4. Each node returns the approximation of the corresponding part of the likelihood.
5. The master node recomputes q by combining the messages received with the prior.

The result is the **same** as in EP but the cost is $\mathcal{O}(m^3) + \mathcal{O}(n/Km^2)$.

Very big gains in the case that $n \gg m$.

The computation of the **gradient** can be distributed in a similar way.

EP Algorithm with Stochastic Gradients

The gradient of $\log Z_q$ contains a **sum** over the data instances:

$$\frac{\partial \log Z_q}{\partial \xi_j} \approx \eta^{\mathrm{T}} \frac{\partial \theta_{\text{prior}}}{\partial \xi_j} - \eta_{\text{prior}}^{\mathrm{T}} \frac{\partial \theta_{\text{prior}}}{\partial \xi_j} + \frac{n}{|\mathcal{M}_k|} \sum_{i \in \mathcal{M}_k} \frac{\partial \log Z_i}{\partial \xi_j}$$

where $\mathcal{M}_k$ is a mini-batch. **Allows for more frequent updates!**

We update hyper-parameters after processing each mini-batch!

Detailed EP algorithm:

1. Process mini-batch $\mathcal{M}_k$ by updating each $\tilde{\phi}_i$ so that $i \in \mathcal{M}_k$.
2. Reconstruct the posterior approximation q.
3. Compute a noisy estimate of the gradient of $\log Z_q$ w.r.t. each ξ_j.
4. Update all model hyper-parameters ξ_j.
5. Reconstruct the posterior approximation q.

If $|\mathcal{M}_k| \leq m$ the cost is $\mathcal{O}(m^3)$ in time and $\mathcal{O}(nm)$ in memory.

Experiments on Datasets from the UCI Repository

Methods compared: scalable expectation propagation (SEP), the generalized FITC approx. and scalable variational inference (SVI).

Average negative test log likelihood across 20 splits of the data.

	$m = 15\%$			$m = 25\%$			$m = 50\%$		
Problem	**FITC**	**SEP**	**SVI**	**FITC**	**SEP**	**SVI**	**FITC**	**SEP**	**SVI**
australian	0.678	0.694	**0.627**	0.683	0.666	**0.626**	0.673	0.637	**0.627**
breast	**0.101**	0.110	0.102	0.111	0.113	**0.103**	0.106	0.113	**0.101**
crabs	0.066	**0.062**	0.068	**0.063**	0.064	0.073	**0.061**	0.062	0.090
heart	0.427	0.402	**0.394**	0.421	0.407	**0.395**	0.416	0.406	**0.396**
ionosphere	0.298	0.264	**0.260**	0.292	**0.272**	0.273	0.302	0.270	**0.257**
pima	0.535	0.524	**0.492**	0.533	0.509	**0.496**	0.528	0.499	**0.491**
sonar	0.354	**0.331**	0.401	0.348	**0.318**	0.404	0.349	**0.290**	0.345

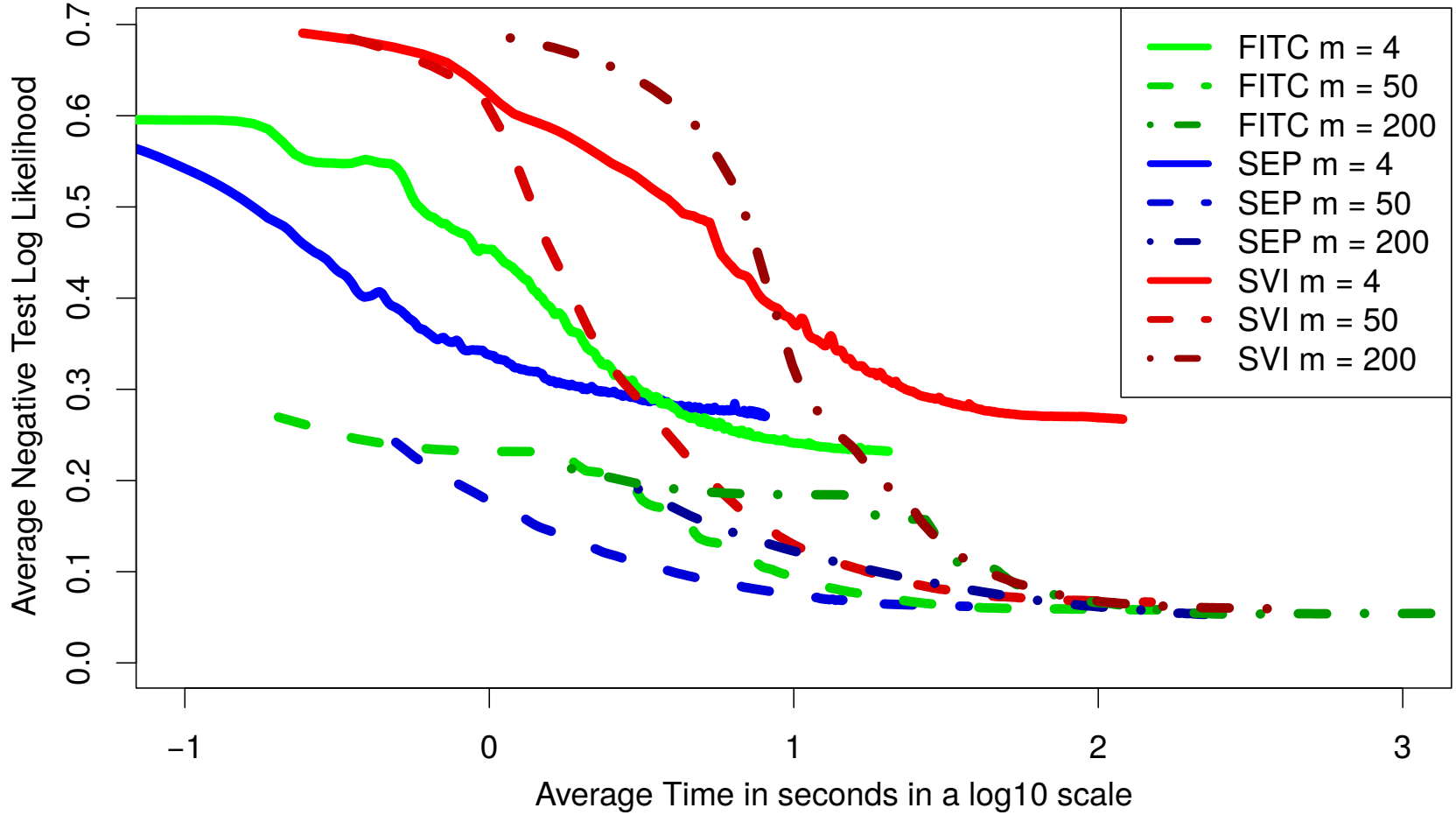
Average training time in seconds.

	$m = 15\%$			$m = 25\%$			$m = 50\%$		
	FITC	**SEP**	**SVI**	**FITC**	**SEP**	**SVI**	**FITC**	**SEP**	**SVI**
Time	58.6	17.3	39.7	132.6	37.1	64.6	493.6	129.6	194.6

All methods are batch (trained for 250 iterations) (FITC and SVI use L-BFGS).

Performance as a Function of Time

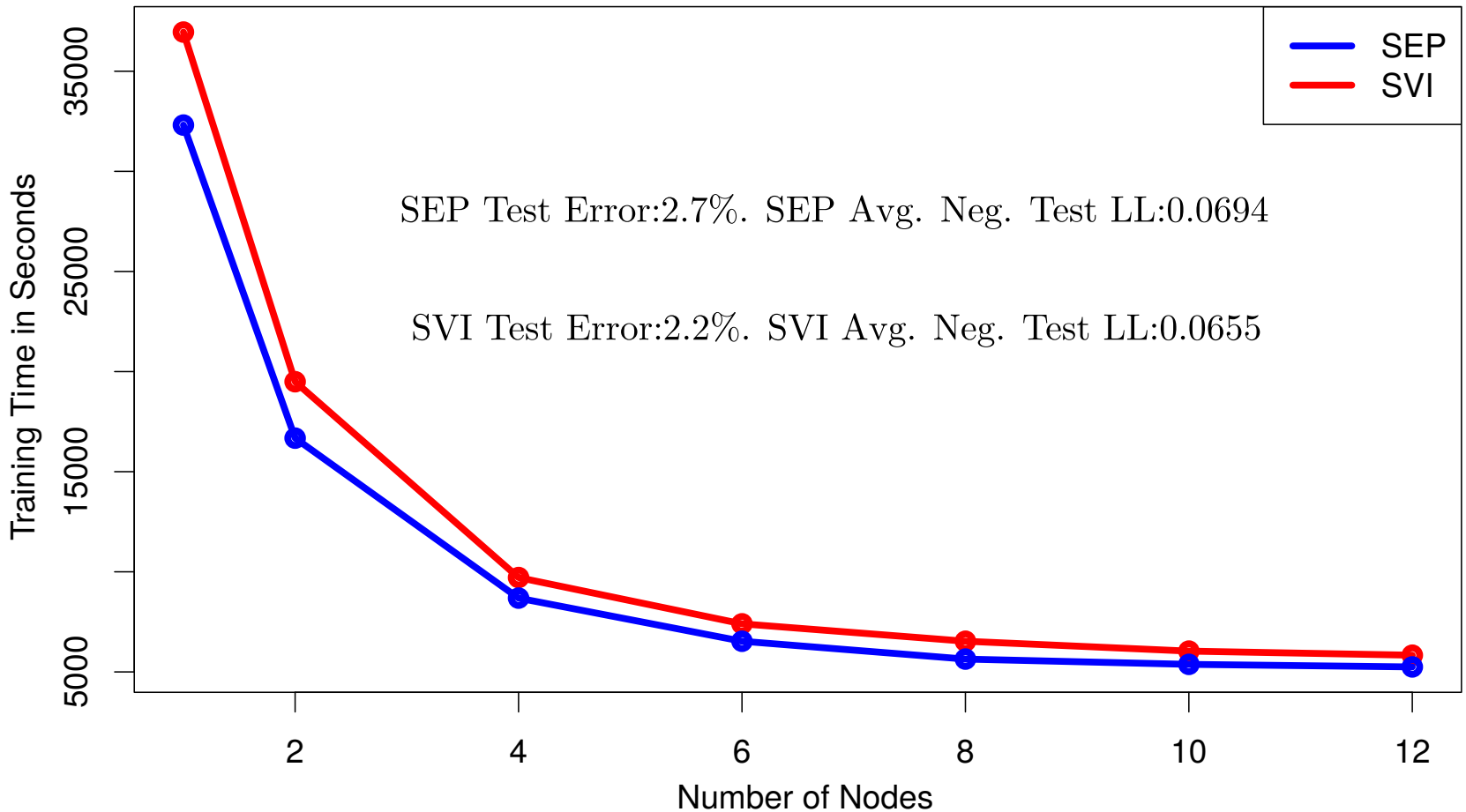
Image dataset which contains 2,310 instances.



All methods are batch (trained for 250 iterations) (FITC and SVI use L-BFGS).

Computational Time w.r.t the Number of Nodes

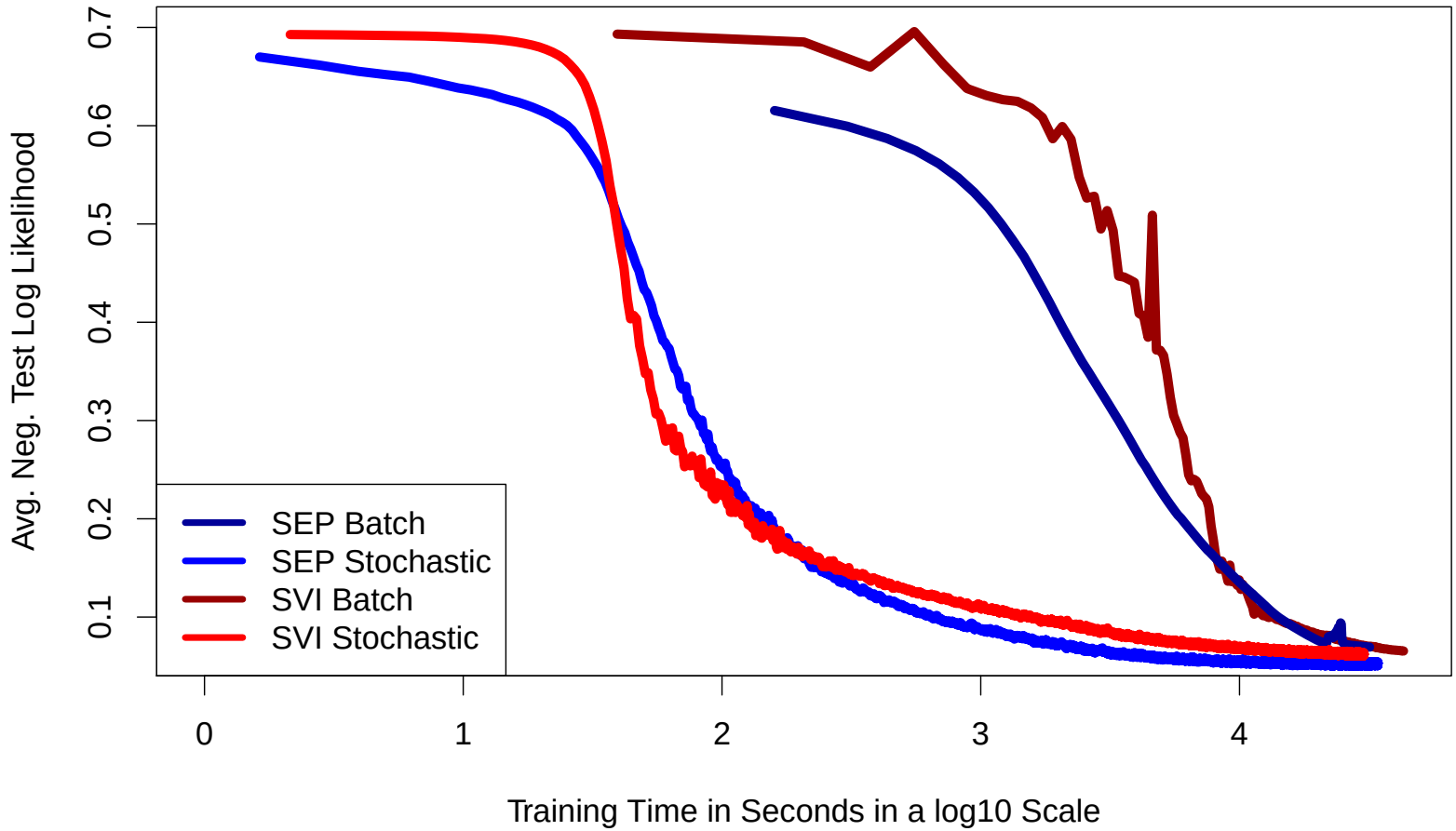
MNIST: 60,000 training instances. Odd vs even digits. $m = 200$.



Parallelization via doMC R package. After 6 nodes there is no improvement. Processes synchronization becomes a bottle-neck. All methods are batch (trained for 250 iterations) (SVI uses L-BFGS).

Stochastic Gradient Optimization I

MNIST: 60,000 instances. Odd vs even digits. $m = 200$. $|\mathcal{M}_k| = 200$.

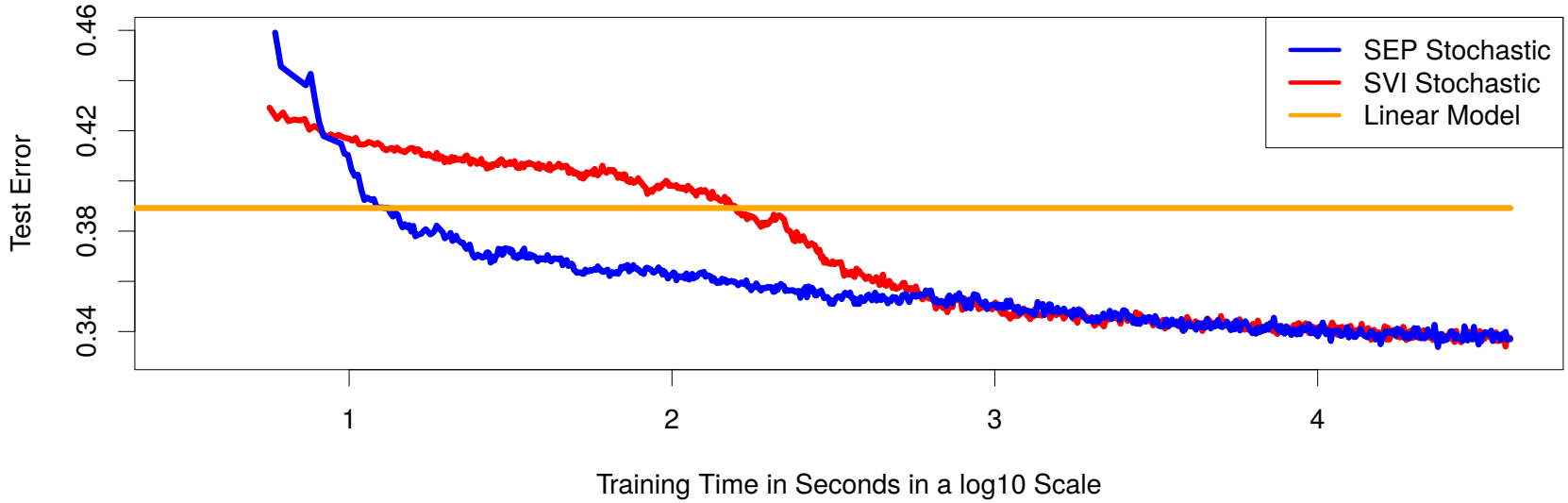


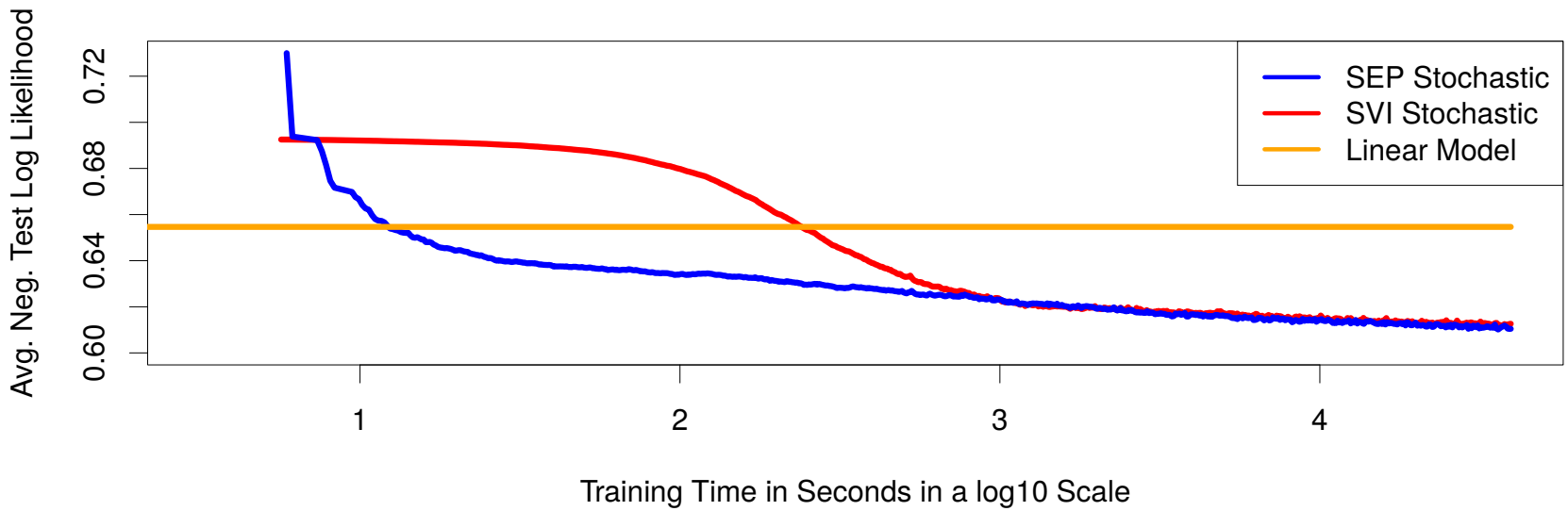
Stochastic methods provide good results before batch methods take a single step.

Learning rates are estimated using the ADADELTA method (Zeiler, 2012).

Stochastic Gradient Optimization II

Airline dataset: **2,127,068 instances** (flights between Jan, 2008 and April, 2008). Predict if there is delay using 8 attributes. $m = |\mathcal{M}_k| = 200$.





Instances with missing values are not considered.

Conclusions

- We have shown that expectation propagation (EP) can be used to efficiently train Gaussian Process Classifiers.
- The proposed method can be used for distributed training and for computing a stochastic estimate of the gradient.
- The proposed method is competitive with related methods based on variational inference (Hensman et al., 2015).
- All updates have a closed form and quadrature techniques are not needed in practice, unlike in the SVI method.
- The training cost of the method is $\mathcal{O}(m^3)$. However, a disadvantage is that the memory requirements are $\mathcal{O}(nm)$.

Thank you for your attention!

References

Gelman, Andrew, Vehtari, Aki, Jylänki, Pasi, Robert, Christian, Chopin, Nicolas, and Cunningham, John P. Expectation propagation as a way of life. *ArXiv e-prints*, 2014. arXiv:1412.4869.

Hensman, James, Matthews, Alexander, and Ghahramani, Zoubin. Scalable variational gaussian process classification. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015.

Heskes, Tom and Zoeter, Onno. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 216–223, 2002.

Hoffman, Matthew D., Blei, David M., Wang, Chong, and Paisley, John. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

Minka, T. Expectation propagation for approximate Bayesian inference. In *Annual Conference on Uncertainty in Artificial Intelligence*, pp. 362–36, 2001.

Naish-Guzman, Andrew and Holden, Sean. The generalized fitc approximation. In *Advances in Neural Information Processing Systems 20*, pp. 1057–1064. 2008.

Qi, Yuan (Alan), Abdel-Gawad, Ahmed H., and Minka, Thomas P. Sparse-posterior gaussian processes for general likelihoods. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 450–457, 2010.

Quiñonero Candela, J. and Rasmussen, C.E. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, pp. 1935–1959, 2005.

Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. ISBN 026218253X.

Seeger, M. Expectation propagation for exponential families. Technical report, Department of EECS, University of California, Berkeley, 2006.

Titsias, Michalis. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

Zeiler, Matthew D. Adadelta: An adaptive learning rate method. *ArXiv e-prints*, 2012. arXiv:1212.5701.